NGR-33-008-090

STATE-OF-THE-ART ORIENTED REVIEW OF CIRCUS

T.E. Stern and H.D. Eskin

Columbia University, N.Y.

CIRCUS is a digital computer program for time domain (transient) analysis of circuits containing linear R's, L's, and C's, independent current and voltage sources and certain semiconductor elements: transistors, diodes, tunnel diodes and four-region devices. The semiconductor devices are represented by built-in models, (the Beaufoy-Sparkes model for transistors) and include photocurrent generators for use in radiation effects studies. Device parameters may be specified by the user, some in the form of tables: a slight departure from similar programs based on model libraries. In overall structure this program is similar in many ways to NET-1. However, it incorporates certain variations which are worth discussing in some detail.

We consider below, the following basic mathematical operations: a) formulation of the network equations, b) computation of initial conditions, c) integration of the differential equations. (I/O features will be omitted in this review, being more pertinent to a user-oriented discussion.)

a) The by now standard procedure of Bryant is used in a modified form (due to Wilson and Massena)(1), to set up the network equations. This permits the analysis of networks containing loops of capacitors and voltage sources, as well as cutsets of inductors and current sources. an improvement over certain other programs which avoid this topological problem by requiring the user to associate a passive element with each source. The fact that the linear elements are sorted according to magnitudes of element values improves the numerical accuracy of the matrix operations required in setting up the network equations. Difficulties may still arise, however, if extremely large ratios of element values occur.

b) A modified Newton-Raphson iteration is used to determine initial conditions. Since this requires a knowledge of derivatives of the nonlinear element characteristics, it is difficult to employ correctly in programs accepting general user-defined nonlinear elements, but well-suited to programs of this type, restricted to built-in nonlinear device models. Nevertheless, the fact that certain device parameters are inserted in tabular form can cause trouble in CIRCUS, since at certain points in the computation it treats the tabulated parameters as if they were constants rather than variables. As a result, an unfortunate choice of tabulated parameter values can cause lack of convergence of the iterations. (We have verified this with examples.) From a practical point of view, however, this is a minor problem since the iterations will converge quite rapidly for most physically reasonable combinations of parameters, and even when they do not converge, the transient computation will often proceed without difficulty.

c) The most interesting feature of this program is the exponential integration routine of D.A. Pope [2] which, to our knowledge, is new to network analysis programs. It is based on linearizing the differential equations over each integration time interval and then evaluating the solution of the linearized equations (exactly) in terms of a matrix exponential. As in the case of the Newton-Raphson procedure, this method requires a knowledge of the derivatives of the nonlinear characteristics and therefore would not be suited to programs accepting more general types of nonlinearities. By using certain short cuts, computation of the exponential (as the sum of a vector series) is accomplished in a highly efficient manner. The value of the shortest (local) time constant is computed as a by-product of the evaluation of the matrix exponential, and this is used in estimating the step-size in an optimum fashion (as large a time step as possible without producing an excessive number of terms in the exponential series). Despite the fact that a series is evaluated at each step, this method of integration and step-size control produces a relatively rapid (noticeably faster than many typical methods) and accurate result, which is especially effective in linear and slightly nonlinear problems. However it is not without flaws. The complete step-size control algorithm involves a number of factors too numerous to discuss here, but their net effect is to cause some serious problems in computing growing exponential solutions. (For example, if one attempts to analyze a linear circuit containing some negative elements, the step-size control may cease to function properly.) This is something that could easily be remedied by a slight modification of the program. Finally, it is worth noting that the integration routine stood up fairly well to the task of computing solutions with large ratios of time constants: a rigorous test of any integration routine. (In our tests it handled ratios up to 10^5 before trouble began.)

In summary, there are no serious deficiencies ... the mathematical procedures, and there are several worthwhile innovations.

Since CIRCUS is based on a built-in model library, it has the basic and serious disadvantages associated with such a restriction: Limited flexibility, generality, and growth potential. The other side of the coin is that the restriction on device models enables one to increase the efficiency of the computation and reduces the likelihood of obtaining anomalous results. We were impressed by the extent to which knowledge of the structure of the device models was used for such purposes as simplifying the computation and speeding up convergence. However, this intimate relation between model structure and program structure suggests that expansion of the program to any significant degree would be rather difficult. These remarks are simply a statement of the fact that there is a definite trade-off in circuit analysis programs between generality and efficiency. This program clearly sacrifices the former for the latter, and if this is what the user is after, he should find it a fairly reliable tool.

One final remark. Although there are certain lapses (notably in the description of the step-size routine) the documentation for this program is by far the best we have seen. ... cular, the complete Fortran listing enabled us to track down several points which would otherwise have been obscured. It will also serve the adventurous user as a starting point for program modifications.

References

1. Wilson, R.L., and W.A. Massena, "An Extension of Bryant-Bashkow A Matrix," IEEE Trans. on Circuit Theory, Vol. CT-12, pp. 120-122, March, 1965.

2. Pope, D.A.,"An Exponential Method of Numerical Integration of Ordinary Differential Equations," Comm. ACM, Vol. 6 No. 8, pp. 491-493, Aug.1963.